Welcome everybody. Today we will give you an introduction to EAC-CPF Version 2.

Hosted by the Society of American Archivists and most specifically by the Technical Subcommittee on Encoded Archive Standards.

Your speakers today are me, Karin Brendenberg, co-chair of TS-EAS, and also working at Kommunalförbundet Sydarkivera in Sweden.

With me, I have Marie Elia, who is the team lead of EAC-CPF, coming from University at Buffalo in the United States, and we also have Ailie Smith, one of the team members coming from the University of Melbourne in Australia.

So I will kick off and give you an introduction to the all over things, and then Marie and Ailie will introduce you to all the new features of EAC-CPF Version 2.

So TS-EAS. I think you heard it before, we are the Technical Sub Committee on Encoded Archives Standards. That's why we have an acronym. So the background to our work is covered in an earlier presentation which is also available on Youtube.

The presentation that we are using today is available already on Github. If you want to have the links to work, you need to download it, but it's already there for you, so you can when we have finished, start looking at it.

Well of course, a part of why we don't provide all the links in the chat is that we are available in a lot of places. So please take a look in the presentation, and see where you can find us.

When it comes to the standards we are maintaining, we do have revisions. We have an annual rolling revision cycle for minor releases. And things happen every year, see more on the tab where we have described this, but every fifth year we need to look at our standards and see if we need a major revision, and that is what has happened with EAC-CPF. So there we are following the guidelines by the Standards committee. So you can read more about that also on the web.

What we also are following working in TS-EAS is the design principles. There are ten, and I will only highlight the first one, because you will see that is a thing that has come through in the whole of this revision of EAC-CPF.

Simplicity comes first. So have that in mind, we are making everything easier. So rest of the schema principles, and to get you all on the same page… I don't know how familiar all of you are with the EAC-CPFs… I'm reusing some content developed from us.

Short introduction to the TS-EAS standard standards that are available on the Youtube channel and short short, what is encoded Archival, context, corporate bodies, person and families. The

goal is to describe corporate bodies, person and families related to our current material; and a corporate body is everything that is not a person and a family.

So society, a small company, and so on. Everything that is not a person in the family. So we have had the international standard on archival authority records being the base for the work.

So we are, EAS-CPF is the technical representation of that standard; so everything is familiar if you have read ISAAR-CPF.

We also quite quickly will end up in XML. So XML stands for Extensible Markup language, I think you'll know that already, and we have the elements, throughout the presentation.

If something is in brackets, it's an element, and if it has the @ sign, it's an attribute. So element contains attributes, and we have elements, values, and we have attribute values.

You also need to remember that we have parents and children. So parent elements with child elements, and that can be going on for a long time since it's possible to make it hierarchical.

XML schema is something that we will also hear more about. There is where we define which elements in attributes that can be used or have been used.

Data type. The data type is telling what the element and method attributes can contain is a text, numbers text following specific rules, and so on. And the last thing that we also will throw you into is namespaces.

So this home space for elements and attributes in the schema. Where do things belong?

Here we are already in the XML. So the structure for EAC is, you have it, the <eac> Root element, and then you have <control>, which is the information about the XML file. You will hear a lot more about that later on, so I'm not going to stop there and give you all that.

We are going to look into CPF description, and where we actually describe the entity so that one has three child elements, the <identity>, where we actually describe what type of entity we are describing.

Identifiers, the name and the name in all its forms.So really the first basic thing, so identifying our entity, the second one is the <description> element where you actually give the description. It's here where you can give all the information you need to give about this entity, existence, functions or occupations, the graphical or historical notes, places, you name it, it’s there.

Also, these elements are…some are more, you can write a really long narrative, or it's more intake intended for indexing. So both of these types of descriptions are there, and you just write along and describe your entity.

A short example, where you can see that we both have the descriptive textual parts, and we have more indexable parts.

The last part that we describe are relations, so every entity we are describing has a relation to something. Either another corporate, body, person, or family. We can describe the relation to other archival resources, or we can describe to functions or whatever so what we really need to create a relation to.

A short example is this buildup of how a family is constituted, where we have a grandfather, a mother, and a father. So everything here is created… this family tree is created through the relationships…Relations or area. And we have these relations to actually make sure that we can connect everything, since archival material, they are kind of complex, and we need to be able to link and show how things are connected.

So that's why we have the relations. When we have all this information, we need to create the EAC-CPF documents in some way; and you can either do this through an XML editor someone has helped you with writing a script so the information is gathered from other structured descriptions, or you are lucky enough to have a collection management system that actually can create these files for you.

When we have the five we want to show them. Of course, these are made for sharing, so you can actually use a stylesheet to transform the XML into HTML displays. You might have a collection management system, that is, having a public interface, so that patrons actually can search and retrieve from it; or you might have a special bill system for showing off the EAC-CPF documents.

Free places are where you can actually look at EAC-CPF live, if in the “Connecting the Dots: Samuel Johnson and His Circle”, which was a project in 2012, where EAC-CPF creates and shows all the connections for Samuel Johnson. SNAC is something really familiar for you in the U.S. and there you also have connections in EAC-CPF. It's built upon the EAC-CPF.

Coming over here to Europe. We have Archives Portal Europe, which also shows and use EAC-CPF for the creators. And with that I will actually hand it over to Marie and keep quiet, and let you tell us everybody what has been done during the last couple of years.

So the focus of today's Webinar is on the updates and the content. So I'm just going to briefly summarize the revision process here.

For context. We're just over 10 years but 12 years since the end of the original version of EAC-CPF.

And this is the first major revision which the team began working on, in 2017, releasing a minor

update. The following year, and placing the call for comments.

So the resulting overhaul at the standard was submitted to the Standards Committee and the SAA Council earlier this year. EAC-CPF 2.0 was adopted in August.

The overall goals for this major revision were simplifying where possible, aligning with EAD were useful, implementing features and solutions based on user requests, and cleaning up any unused components.

So you can get a very detailed explanation of the revision process on the EAC-CPF Website. And if you go to the page on the sign on the sidebar called EAC-CPF 2.0 background and then click on the revision notes.You'll see a long, detailed explanation of the whole process.

But now I'm gonna give you an overview of some of these changes before we get into the details.

The goal of EAD alignment is to better meet the user community's needs and allow the standards to work together. So some elements were renamed as part of the alignment, and some were renamed to be more precise. Ideally these changes will make the schema easier to use and understand.

Also as part of the alignment, some elements were removed or replaced, so I know this is a lot of text on the slide, but you can see sometimes elements were replaced by a new equivalent term or different term, and sometimes by a combination of element with an attribute value.

Also we're getting simplification. EAC-CPF 2.0 bundles elements in these two ways So elements of the same type can be bundled with a wrapper element. Wrapper as plural element and elements with different concepts can be grouped in element sets.

There is a prescribed order for elements within the parent elements,so prioritizing, required and non-repeatable elements.

Restrictions were relaxed for some element content and attribute values, and you can see some examples here. Agency code. The ISO constraint is relaxed, and for country language and script code, the ISO standards constraint is relaxed to name token. And if you're not using XML a lot, the point here is that there's more flexibility for values.

I mentioned a couple more things, but Ailie is going to be covering some of these in more detail, including control in general, as well as showing examples of encoding. So briefly in <control> these new attributes align with EAD, so encoding for country, date, language, repository, and script. As do these new optional elements.

The removal and replacement of attributes regarding external namespaces, also further align

with EAD.

And finally, there are five newer replacement global attributes, and we will have a detailed explanation of both internal and external referencing later again with examples. So that is a brief overview, and we'll move to Ailie to go into Control.

Thanks, Marie, so as part of the revision process that we looked at making a number of improvements to the control section of the scheme. So that's where information about the actual EAC-CPF document is encoded

So elements that have been used to encode the status of a record. So things like maintenance, status and publication status have been transformed from elements within control, into attributes of the control elements. These attributes continue to use the same limited set of values that were available for the elements in the previous version of EAC-CPF but the terms are currently on discussion as well.

Other elements that have been transformed into attributes in disposed to EAC-CPF elements that contain information about the type, so maintenance event type has become an attribute of the <maintenanceEvent> element, and agent type has become an attribute of the <agent> elements; as with the status attributes, these continue to use the same set of values that these elements had in the previous version with the attributes in the current version.

So several elements in the control section have been enhanced to improve some of the functionality. Previously you could encode conventions, used sources, maintenance, events, and local types in the control section but it wasn't very easy to connect them with descriptive details within the EAC-CPF documents.

So now, by including an ID attribute on these elements, you can refer to them from the other elements in the record by using the relevant reference attribute.

So if you wanted to, for example, connect data first to the source where the information was found, you can encode the source with an ID attribute in the control section of the EAC-CPF record, then include source reference on the start date element to refer back to that ID. And there will be some more information about using references within an EAC reference.

So I'm going to go through just a couple of use cases for control.

So in this case we're looking at creating a new EAC record with information about a working group who created it and when it was created. So we have a unique ID, an agency name, and a bit of information about the creation of the record.

And this is the example encoding that that can produce. In this case, this is about the minimum information that you can have within a control record. So you can see that using the

maintenance status attribute there, we know it's a new record, and the publication status indicates it's in progress.

We have a unique identifier. The agency name. And further down we can also see that each the agent name is included there. The record is created with the agent type human, so we know it was created by a person, it's not machine generated, and we have the date there, that this record was created.

So building on this previous example, we can start to add further details. So the Creator here has identified that it does have this record as an external audience. The publication is still in progress, and that they wanting to use some data standards to create their record and want to know how to encourage that in the record.

So we have the various different encoding attributes there, to be used to encode the countries, dates, languages, repositories, and scripts.

And so adding all of these attributes, you get a control elements that looks like this.

You can see the maintenance, status, and publication status that were in the previous example they highlighted there, and these are now joined by the audience attributes stating that it's an external audience, and all the encoding standards for the record there.That's the different ISO standards.

So moving on to another use case, and this time we're looking at the maintenance event element specifically, which is within control.

In this case the user wants to make an internal note about what they did and why.

So we still have the name and the date type information. But we also have a description of what happened in this maintenance event.

And so here we can see how this looks in the actual XML encoding. We have a maintenance event element with the type created. We have the agent type, human again, so it's not a machine generated record.

We have the date that this record is created, both in text and in a standardized form on the standard date time attribute, and in this case we have the event description, so example for webinar. So this tells us that this particular record was created to be used as an example for this webinar.

Linking and referencing, so moving away from control, we're going to talk about how linking and referencing works in EAC-CPF 2.0.

So there's a number of options for external referencing within the schema. It's possible to reference external vocabulary ontologies, using the attributes value: URI vocabulary, source and vocabulary source, URI. And these are available on a large number of elements within the schema, and so these can be used to point to external references where terms and vocabulary are being drawn from.

It's possible to reference external sources for the contents of your EAC-CPF record, using the reference element which sits within source in control. And this can have an @href to provide a URI to a specific web page, or it can also just include text.

The reference element is also available within several other elements within the record, including the events and abstract to reference externals sources for some context.

So here we have a few different examples of using external referencing at the top piece of code, you can see that this is a source. This would sit within control.

The source is the Barack Obama Presidential Library, and we have an @href that points you directly to that reference. In the second example, there we have an elementary element.

This is encoding the name Barack Obama, and this name has been taken from the vocabulary source from Wiki data, so you have the vocabulary source URI there. And you have the value URI that points to the specific name within this.

And in the bottom section we have a relation, so is the relation elements that is pointing to a corporate body, so the Democratic party. At the bottom we have a reference that provides more context for this relation with a URI there as well.

So there are also some enhanced options for creating internal references within a single EAC-CPF document.

So the ID attribute is available on all elements to assign and identify that is unique within the instance to those elements. There's also a new target attribute that can be used to refer to the ID of another element from one element, so that's the way you can create links between elements within a document. You can also reference specific elements in control from some of the descriptive elements in the EAC-CPF instance using specific attributes.

So I'll be talking a little bit more about this in the next section, but first we can have a look at using ID and target attributes to create some internal referencing.

In this example, here we have an occupation element at the top, so the occupation is an Assistant Examiner and then we have the place name, and the place name has a target pointing to an identifier. So this identifier place1 you can see below is the ID of the place element.

And here we have a more detailed description of them as a place, including an address and the role of that place within its record, is a place at work. So using that ID on place and the target on place name, means we can connect those details together so you don't have to describe in great detail, if every time you can just link back to the place there.

So based on some use of feedback. This version of EAC includes the ability to encode evidence-based assertions.

For statements for part of the EAC-CPF description, you can encourage who added the assertions based on which source and following which rules. And this is especially useful, where there might be conflicting statements in the EAC-CPF description, such as different spellings of a name or different dates of birth that have come from different sources.

So the new maintenance event reference attribute at source reference and convention declaration attributes allow you to refer from a descriptive element in the EAC-CPF instance, back to maintenance event source and convention declaration elements in the control section. And you can reference more than one of these elements to encode this information.

So here we have an example of how this is working. So I'm going to start by looking at the code on the bottom of the screen, which is a relation entry for Princeton University.

It has a source reference, source1 and a maintenance events reference, me2 there, which would refer back to the IDs of things in the control section. So above you can see that source1 is the ID of source and this is the history of the University of Oregon.

And the maintenance event is the top thing there, so the maintenance event was updated by John Smith in 2021. So we know that this relation was added by John Smith in 2021 with the source space there of the history of the University of Oregon, specifically page 270, so we know exactly how that statement in the relation element came about.

So local types. Local types continue to be used in this version of EAC-CPF as an attribute on elements, and you can have references in control for the local tag declaration, where you declare what these local types are. There is now a local type declaration reference attribute that can be added that can connect the two of those.

So here at the bottom we have a name entry for Hannah Arendt and we have a local type; this is a personal name and the local type declaration references GNDO. So going back up to the top piece of code there, which would sit in the control section, local type declaration with GNDO shows us that this local type is drawn from the GND Ontology.

So now I'm gonna talk more about the descriptive elements that have had some changes made in EAC-CPF version 2.0 and we'll start with encoding of names.

So there've been several changes made to names within this version, so name entry parallel has been transformed into name entry set and this is a rapid element for grouping two or more name entries that represent different forms of the same name. So, for example, the same name in different languages or different scripts.

If you wish to indicate that this is a parallel name, you can use a local type with the attribute, with the value parallel on the name entry element to continue to do this. Once again there's several elements related to names that have been transformed into attributes. So, instead of having authorized form and alternative form elements, there's now a status attribute that can have the values alternative or authorized to continue this functionality. And instead of having a separate preferred form element there is now a preferred form attribute available for name form entry and this can have the values true or false.

So if you wanted to indicate that one particular main entry is the preferred version of a name, you can use that preferred form attribute on the name entry with the value true.

And moving on to some examples. So here at the bottom of the screen we have a name entry has a preferred form true, this is the preferred form of the name. The status authorized, and the convention declaration reference of cd1. And now this convention declaration reference points back to the convention declaration and control with the ID cd1, and this is the reference ethanol, and it's in Z44-060, and that is the conventions that's been used to formulate this name.

So in this example we have a name entry set with a local chat parallel, and this example shows two forms of the same name.

So the first name entry is the German version of the name indicated by the language of element, and the second version is a Japanese version of the name and uses a Japanese script and you can see this in the script of element attribute there. Looking at some of the other attributes of the name you can see that while they're both authorized versions of Hannah's name, the German version is the preferred version and local type attributes have been used to indicate the first name entry is the native German version of Hannah's name, and the second name entry is a Japanese translation of the name.

So places can be fully encoded within a place element in this version, and be multiple places within a plural places proper element. Place can also sit within relations and form part of structured chronological lists of events.

In this version place entry has been renamed as place name; so individual place names can be included within singular elements, such as function, occupation, and mandate without a place wrapper element, and the associated encoding.

So you can include a lot of context on places in this version. So the place element requires at least one of the name of the place, place role, the physical address, digital contact information

and geographic coordinates. So place name is highly recommended to be included, but it must include at least one of these.

In addition to this version of EAC-CPF is the new contact element. This works very much like the address element that existed in the previous version, but where address encodes a physical address, contact can be used for providing contact details such as email addresses, websites, phone numbers… those sorts of things. Geographic coordinates has also been added as an element in this version to align with EAD, and previously you could encode it using attributes, but now, with these elements of its own right.

You can also encode date, information, and other descriptive information within place to give further context.

Example. So here we have a place with a lot of detail, so we know the place name is the Tokyo Imperial Palace. It has a place role, so it's a place of birth for the person that this record is about. We have geographic coordinates. We have a complete street address, and we also have contact details with the homepage for the Imperial Palace included there.

Okay. So there's also been some changes to the way dates can be encoded, and this has been based on alignment with EAD, and with some user feedback that based on the need to include information about uncertain and unknown dates in EAC-CPF.

So to achieve this EAC has adopted the certainty attribute from EAD as well as the calendar and era attributes in order to give more detail about dates. EAC version 2 also enables the use of the extended date time format, which can be used for encoding things like certainty and continuing date ranges.

And this has been included in latest ISO 8601 standard, and you can now use that within EAC-CPF.

A new status attribute has also been added from date and to date to allow for including information about where a date range or part of a date range may be unknown or where a date range is still going, so it doesn't have an end date.

These are a few examples of some dates. In the first example, we just have a single date. This includes a certainty attribute with circa, so we know that the date is 1789, not exactly 1789.

The second example is a date range. This date range we know the end date, which is circa, 2010 and that is also using in the standard date, the extended date time format. So that's what the question mark on the end is encoding. However, we don't know the start date of this range, so the from date has a status unknown.

In the third example, we have a date set which includes the singular date, July 2014, but we

also include a date range that is on ongoing so the date range hasn't ended there is no to date so the to date has the status ongoing.

So another section of EAC-CPF that has had a number of changes and new element attributes added is the relations section.

So, instead of having separate elements for function, relations, resource, relations, and relations to other CPF entities, there's now a single relations element that that covers all of these. So every relation element must include a target entity element with a target type attribute, so the target entity is where you identify the relation entity and the target type attribute is used to identify what the target entity is that can be an agent, corporate, body, family, function, personal resource.

So instead of say, having a CPF relation, as you would have previously, you would now just use relation with the target entity with that includes the name of that entity, and that tag entity would have an attribute, type could be person, for CPF, could also be family, corporate body or agent, and the same works for functions and resources.

We also have several new attributes within relations, so we have a new relation type. And this specifies the type of relationship that's being described that the target entity has to the targeted entity.

We also have target role that can be used to provide information about the role of the target entity towards the entity described, and we can use further details from date, date range, date set, place, descriptive notes and object XML wrap to include XML from somewhere else that can provide for the context about a relationship.

So here is an example of a relation entities, So in relations it's actually wrapping multiple different relations there. So we have a full relation entity described here. The target entity has target type person.

We know the name of the person is Paul Arendt, and we're giving a relationship type of family and a target role of parents. So Paul is the parent of the person who is actually in the EAC-CPF instance that we're looking at here.

Well I may pass this over for some talk about documentation.

Thank you, and before I start I'm actually going to put all of these links in the chat right now.

So you can see them, but they are also available in the PDF of the slides that we linked to earlier.

So this is just an overview of highlights and changes we think will be most noticeable, and

hopefully most useful to the user community. But we encourage you to look at the EAC-CPF website which has more detailed information and there you'll find again, details of the revision, a link to the Tag library, which does include snippets of examples, for how that element or attributes are used.

There's also a link to the TS-EAS Github site, where you'll find the new best practice guide and our team member, Iris has created a short term tutorial video which is also linked here, that's on the SAA Youtube channel that gives you a little tour of how the best practice guide works.

So we have shown lots of examples in use cases and on the Github site, there's an example of a brief record that's encoded, and an extended record, as well as to just sort of sample records. But we would like the best practice guide to include your use cases and examples; And you can submit examples or raise issues through the Github site.

You can contact us. We think that the best practice guide will work best. We can actually see people's real world examples of them. So if you're using this and you have questions, or you have examples. We would be very glad to see them

Well we have we have managed to do this in 45 min. Many, many thanks for all of you who came and listened to us.